Partitioning based incremental marginalization algorithm for anonymizing missing data streams

Otgonbayar, Ankbayar; Pervez, Zeeshan; Dahal, Keshav

Published in:
Proceedings of the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)

DOI:
10.1109/SKIMA47702.2019.8982399

Published: 06/02/2020

Document Version
Peer reviewed version

Partitioning based incremental marginalization algorithm for anonymizing missing data streams

1st Ankhbayar Otgonbayar
AVCN, CEPS
University of the West of Scotland
Paisley, United Kingdom
Ankhbayar.Otgonbayar@uws.ac.uk

2nd Zeeshan Pervez
AVCN, CEPS
University of the West of Scotland
Paisley, United Kingdom
Zeeshan.Pervez@uws.ac.uk

3rd Keshav Dahal
AVCN, CEPS
University of the West of Scotland
Paisley, United Kingdom
Keshav.Dahal@uws.ac.uk

***Abstract*—The IoT and its applications are the inseparable part of modern world. IoT is expanding into every corner of the world where internet is available. IoT data streams are utilized by many organizations for research and business. To benefit from these data streams, the data handling party must secure the individuals' privacy. The most common privacy preservation approach is data anonymization. However, IoT data provides missing data streams due to the varying device pool and preferences of individuals and unpredicted devices' malfunctions of IoT. Minimization of missingess and information loss is very important for anonymizing of missing data streams. To achieve this, we introduce *IncrementalPBM (Incremental Partitioning Based Marginalization)* for anonymizing missig data streams. *IncrementalPBM* utilizes time based sliding window for missing data stream anonymization, and it aims to control the number of *QID*s for anonymization while increasing the number of tuples for anonymization. Our experiment on real dataset showed *IncrementalPBM* is effective and efficient for anonymizing missing data streams compared to existing missing data stream anonymization algorithm. *IncrementalPBM* showed significant improvement; 5% to 9% less information loss, 4500 to 6000 more number of re-use anonymization while showing comparable clustering, suppression and runtime.**

***Index Terms*—Anonymization, Internet-Of-Things, Missing data stream, Missing value, Privacy**

I. Introduction

The Internet-Of-Things(*IoT* hereafter) and it's application became the inseparable part of modern world. Constantly expanding big data-sphere is built around the world for the benefit of public and private sectors. Huge amount of sensors, actuators are embedded from everyday items to houses, big buildings thus providing a chance to detect and control precisely. Such as, government agencies, big corporations and research institutions [1] utilize the IoT data for the purpose of research and business. Many organizations utilize IoT data to improve their performance and optimize the business. For example, traffic control agencies use smart car and traffic monitoring data to optimize the traffic control, smart home data are analyzed by housing agencies to improve the house conditions [2], [3].

However, utilizing these versatile data without breaching the personal privacy of the individuals is the primary concern [4], [5]. The privacy of each individual is violated when their identity is exposed to the malicious party, the attackers can exploit this information to learn their personal information by combining with other disclosed data [6]. Therefore, it is mandatory to sanitize IoT data upon handling [4], and the most popular data sanitization approach is anonymization [6]. This method creates privacy barrier on the data by removing or replacing information which can be utilized by malicious party to breach the privacy of individuals, therefore, ensuring the individuals' information are protected from privacy disclosure while publishing [7]–[10]. Identifier of confidential information of individual that poses privacy treat when exposed to the public is called sensitive information. For example, social security number, bank details, address and phone number. In contrast, non-sensitive information that might be exploited by malicious party to create privacy breach is called quasi-identifiers (*QID* hereafter). For example, in health-care data; age, height, gender, ethnicity and area code etc.

The anonymization replaces or removes the value of quasi-identifiers and creates uncertainty provide the privacy preservation [10]–[12]. Anonymization quality is measured by information loss which is calculated by measuring the level of uncertainty in the published information [13]. There are two types of data anonymization; dataset anonymization and data stream anonymization [14]–[18]. Dataset anonymization is performed on a previously recorded data, the main goal this type of anonymization is to minimize the information loss while publishing under user-specified privacy settings [19], [20]. On the other hand, data streams are received in a sequential order and it is processed dynamically [21]. The goal of anonymizing data stream is to publish data stream without exceeding the specified publication delay while maintaining minimum information loss [22]–[24]. However, data streams anonymization quality is measured on the trade-off between publication delay and information loss [25], [26].

Moreover, data streams must be anonymized in dynamic framework with fast and consistent publication. The most suitable technique to handle fast flowing data streams is the sliding window [14]–[18]. There are two type of sliding window; count-based [14], [16] and time-based [15], [17], [27], [28]. Count-based sliding window is size constrained, and data is anonymized and published when sliding window is full. In contrast, time-based sliding window constrained by time

978-1-7281-2741-5/19/$31.00 ©2019 IEEE

and every tuple of time-based sliding window has expiration time. Therefore, data are anonymized and published when the oldest tuple of the sliding window expiring. Researchers agreed that time-based sliding window is the most feasible solution for anonymizing data streams due to its time-bounded publication property.

The individuals of IoT possess many devices that have sensing and actuating capabilities, and those devices can be used at any time. Also, there is no globally defined usage pattern for IoT devices. Therefore, IoT produce data streams that have missing values. There are four main cause for missingness in IoT data streams; different device preference of individuals, dissimilar usage pattern, unpredictable environment condition and personal information sharing control.

Anonymization of missing data streams is very challenging [29]–[31], researchers found three major techniques to treat missingness for anonymization. There are imputation, marginalization and partitioning. Firstly, for imputation, missing values are repaired by counterfeit values calculated upon or during anonymization process [32], [33]. Secondly, missing values are ignored and treated as a $NULL$ value [34]. Finally, for partitioning, QIDs of each tuples are used to create multiple non-missing subsets of original data, then, each subset are anonymized individually [35].

The biggest challenge of anonymizing missing data streams is to achieve minimum information loss with minimized delay while dealing with the missing values. The quality of the missing data stream anonymization is affected by the sliding window size and missingness. However, sliding window is designed to run anonymization iteration for each expiring tuples, and more tuples for the anonymization iteration results less information loss. Also, the less number of QIDs in the anonymization iteration less information loss and have minimization effect on missingess. Therefore, we propose missing data stream anonymization algorithm *IncrementalPBM* which aims to minimize QIDs while utilizing maximum possible tuples for the anonymization under time-based sliding window. The general idea is to utilize partitioning to minimize number of QIDs for anonymization and incrementally expand the field of anonymization QIDs for more number of tuples for anonymization iteration. More details of *IncrementalPBM* is explained in Section IV. Experiments on real dataset proved that *IncrementalPBM* is more effective on anonymizing missing data streams compared to existing algorithm, demonstrating less information loss, suppression; comparable runtime and clustering, and more number of re-using.

II. Background

Data stream is received in sequential order and is to be anonymized on the fly. Therefore, researchers implemented dynamic technique called sliding window to perform data stream anonymization. There are two types of sliding window; count-based and time-based. Wang et al., [22] proposed algorithm called *SWAF* that utilizes specialization tree. When tuple is received, it runs top-down greedy approach to find the best K-anonymous cluster for expiring tuple under time based sliding window. Also, in [24] Jianzhong et al., presented another specialization tree based algorithm *SKY*. This algorithm searches the most specific specialization tree node for newly arriving tuple, then it checks if the tree node is eligible to anonymize the newly arrived tuple. Otherwise, the tuple is held until their corresponding specialization tree node accumulates enough tuples to satisfy K-anonymity.

Cao et al., proposed *CASTLE* data stream anonymization algorithm in [16]. *CASTLE* handles data stream under count-based sliding window. Newly arriving tuples are assigned to new or existing cluster depending on the cluster enlargement cost. The cluster enlargement is measured on the gain of information loss that is occurred when adding new tuple to the existing cluster. Moreover, to reduce anonymization cost, *CASTLE* utilizes re-using strategy that uses generalization information of recently published K-anonymous clusters to anonymize expiring tuples.

Hessam and Sylvia proposed a count-based sliding window algorithm for numeric data streams anonymization called *FAANST* [14]. The aim of this algorithm is to improve the data quality. Therefore, they introduced Δ, information loss constraint of clusters. When sliding window is full, *FAANST* uses K-means algorithm to create clusters then it publishes K-anonymous clusters those having less than Δ information loss. Also, it utilizes re-using strategy to minimize the information loss. *FAANST* outperformed *CASTLE* in terms of time complexity and information loss.

Wang et al. [27], found that *CASTLE* [16] created few huge clusters when applied on data streams, causing frequent split operations. The merge and split operations are costly, in terms of time complexity and information loss. To address this issue, they proposed *B-CASTLE*, they set a threshold on cluster size and applying correlation distance to select merging clusters. *B-CASTLE* showed higher quality anonymization in shorter time.

Guo et al., [15] proposed time based sliding window algorithm *FADS* to resolve the issues of *CASTLE*. They found overload of clusters in *CASTLE*. Also, they discussed that complex merge and split functions of *CASTLE* are not important for data stream anonymization since the size of cluster is defined by K. They solved these issues by utilizing time-based sliding window, exploiting re-usable clusters in the memory, and by using *KNN* for anonymization iteration to publish K-anonymous cluster. Adorenke et al., [36] mentioned that the high information loss can occur when data stream is intermittent. They proposed an adaptive buffer resizing anonymization scheme for resource constrained environment to anonymize streaming crime data. They predict the data arrival rate to resize the buffer to minimize the information loss. For the calculation of data arrival rate they utilized Poisson probability model [37] by assuming the data streams as a sequence of events occurred in a fixed time interval.

Missing value is the one of the most challenging topic for data analytics [29], [30], [33], [34]. There are three main problems that missing data causes: creates substantial amount of information bias, makes data handling and analysis

formidable, and inefficiency [38]. Imputation, marginalization and partitioning approaches are predominantly used to address the aforementioned issues.

Imputation: Imputation is the well-known method for repairing missing values in statistical analysis. In this method, pre-calculated counterfeit values replaces the missingness of the data stream. Imputation is classified into two types: single imputation [39], [40] and multiple imputation [41], [42]. Imputation fixes the missing values of the data, however,it amplifies the uncertainty of anonymized data creating more information loss. If the imputation amount of cluster is less compared to its size, then anonymization with imputation results more secure privacy preservation while repairing the missing values.

Marginalization: In marginalization, missing values are treated as *NULL* values, then, utilized as normal QID value [35]. The main issue of the marginalization based approaches are the excessive amount of missing values of published clusters. However, marginalization is the most feasible solution if moderation of missingness is considered in the publication. Marginalization does not add any values to the original data and it makes published data easier to analyze [34].

Partitioning: Dataset with missing values can be divided to several subsets with no missing values, and each subset can be processed by traditional anonymization methods. However, partitioning based approaches are not cost efficient when dataset have an excessive volume of missingness in comparison to its size [18]. Nevertheless, this technique publishes clusters without missing values while ensuring the privacy preservation. Ciglic et al., proposed partitioning based anonymization approach *ANON* [35] for dataset containing *NULL* values, along with three *NULL* value matching scheme.

III. Preliminaries

In tradition data streams, each tuple have same QIDs; whereas, missing data stream might contain tuples that have one ore more missing values. In the following we define the fundamentals of missing data stream and it's anonymization.

Definition 1 (Tuple of IoT data) *Tuple of IoT data is defined as: $t(id_t, Q_t, ts_t)$- where id_t is the identity of an individual, $Q_t = \{q_1, q_2, ..., q_m\}$ is a set of QIDs of a tuple, and ts_t is a time-stamp of the tuple arrival.*

Definition 2 (Missing data stream) *Let Q be the QID's of the data stream, where $Q = \{q_1, q_2, \ldots, q_n\}$.The missing data streams are defined as: $S(id, Q_t, ts)$ where id is the individual's identity, Q_t is the subset of $Q(Q_t \subseteq Q)$ that describes a receiving tuple, and ts is the time-stamp of the tuple.*

Definition 3 (K-Anonymous cluster) *Let $C(Q_c)$ be a cluster C generated out of missing data stream S. If the $C(Q_c)$ contains not less than K number of identities in its composition, then, $C(Q_c)$ is a K-anonymous cluster.*

Definition 4 (Partition on Q_p) *Let P be a set of tuples that shares exact same QIDs set $P(Q_p)$ $=\{t_1(pid_1, Q_p, ts_1)$, $t_2(pid_2, Q_p, ts_2)$,..., $t_z(pid_z, Q_p, ts_z)\}$, then P is a partition created on Q_p.*

Definition 5 (Distance between two tuples) *Let $t_1(pid, Q_1)$ and $t_2(pid, Q_2)$ be the tuple of missing data stream S. The distance of t_1 and t_2 is calculated on the common QIDs of both tuples.*

$$Distance(t_1, t_2) = \frac{\sum_{q_i \in |Q_1 \cap Q_2|} d_i(q_i)}{|Q_1 \cap Q_2|} \tag{1}$$

$$d_i(q_i) = \begin{cases} \frac{|r_{i.1} - r_{i.2}|}{|R_{i.u} - R_{i.l}|} & \text{if } q_i \text{ is numerical} \\ \frac{|leaves(H_i)|-1}{|leaves(DGH_i)|-1} & \text{if } q_i \text{ is categorical} \end{cases} \tag{2}$$

Where $r_{i.1}(r_{i.2})$ is the value of $t_1.q_i(t_2.q_i)$ if q_i is a numeric attribute, H_i is the lowest common ancestor of $t_1.q_i(t_2.q_i)$ with respect to DGH_i.

Clusters generated from missing data streams can contain tuples with different composition of QIDs. Cluster generalization of missing data streams is not similar to traditional cluster generalization. We define the following cluster generalization for such clusters.

Definition 6 (Cluster generalization) *Let $G_j^*(g_1, g_2, ..., g_m)$ be the generalization of cluster $C(Q_j)$. Following calculations are utilized to find the generalization of each QID of Q_j for generalization.*

1) *$g_i = [r_{i.min}, r_{i.max}]$, where $r_{i.min}(r_{i.max})$ is the min(max) value of q_i in cluster C. If q_i is a numerical.*
2) *$g_i = H_{i.lowest}$ where $H_{i.lowest}$ is the lowest common ancestor of the q_i values of the cluster C. If q_i is a categorical.*

Definition 7 (Information loss of tuple) *The information loss of generalizing a tuple $t(pid, Q_t)$ to $G_t(g_1, g_2, ..., g_m)$ is defined as follows:*

$$InfoLoss(t, G_t) = \frac{1}{|G_t|}\Big(\sum_{q_i \in Q_t} Loss(q_i)\Big) \tag{3}$$

Where $Loss(q_i)$ is the attribute information loss q_i caused by the attribute generalization g_i.

$$Loss(q_i) = \begin{cases} \frac{r_{i.u} - r_{i.l}}{R_{i.u} - R_{i.l}} & if\, g_i \in [r_{i.l}, r_{i.u}] \\ \frac{|leaves(H_i)|-1}{|leaves(DGH_i)|-1} & if\, g_i = H \end{cases} \tag{4}$$

Where $[r_{i.l}, r_{i.u}]$ is the numeric domain of a numeric attribute q_i, and DGH_i is the domain generalization hierarchy(DGH) of a categorical attribute q_i, $|leaves(H_i)|$ and $|leaves(DGH_i)|$ represents the size of a tree rooted on H_i and DGH_i respectively.

Definition 8 (Average information loss) *The average information loss for anonymization of the first N tuples of missing data stream defined as follows:*

$$AverageInfoLoss(N) = \frac{1}{N}\sum_{i=1}^{N} InfoLoss(t_i, G_i) \quad (5)$$

IV. Incremental partitioning based marginalization

The details of *IncrementalPBM* are explained in Algorithm 1. *IncrementalPBM* has four parameters; S-missing data stream, *K*-anonymity degree, δ-time based sliding window contraint and ω is the time contraint for re-using K-anonymous clusters. Firstly, *IncrementalPBM* reads tuple *t* from *S*, then, depending on its QID composition, *t* is inserted into partition in Set_p, or new partition (see definition 4) is created on *t* and added to Set_p. Set_p acts as the buffer in $IncrementalPBM()$.

When tuples is expiring, it removes all expired *K*-anonymous re-usable clusters from Set_{kc}, then, $IncrementalPBM()$ checks the size of the buffer. If the size of the buffer is greater or equal to *K* it calls procedure $AnonymizationPBM(t)$ to publication, otherwise, $SuppressAnonymization(t)$ is called. The details of procedure $AnonymizationPBM(t)$ and $SupppressAnonymization(t)$ are shown in Algorithm 2 and Algorithm 3 respectively.

Algorithm 1 $IncrementalPBM(S, K, \delta, \omega)$

```
Let be Set_p a set of partition which will act as buffer, initialized empty.
Let be Set_kc a set of K-anonymous cluster, initialized empty.
while S ≠ NULL do
    Read tuple t_i from S and assign partition of Set_p or create new partition on it.
    if t is expiring according to δ then
        Remove expired K-anonymous clusters' of Set_kc according to ω
        Let t be the expiring tuple
        if (∑_{P_i∈Set_p} |P_i|) ≥ K then
            AnonymizationPBM(t)
        else
            SuppressAnonymization(t)
        end if
    end if
end while
```

$AnonymizationPBM(t)$ is the most important part of our algorithm. Firstly, it tries to find K-anonymous cluster C_{min} from Set_{kc} which covers t with minimum information loss. If C_{min} is found, then, simply, *t* is published with re-use. If *t* is not published by re-use, it creates temporary partitions set S_{pub} and inserts all the partitions that are defined on the QIDs that can be covered by QID of expiring tuple *t*, and anonymization iteration QID is set as *t*'s QID set. Therefore, number of tuples for *KNN* is increased without extending the QID set of anonymization iteration. Then, it checks the size of S_{pub} to ensure enough tuple is accumulated for *KNN*. If there is not enough tuple for *KNN* it finds most similar and biggest partition using Jaccard's similarity [43] to expand QID set of anonymization iteration. Following the incremental expansion of QID set of anonymization iteration, more partitions are added to S_{pub} for anonymization. This incremental process continues until S_{pub} collects enough tuple for *KNN*. When the incremental accumulation complete, it runs *KNN* and forms new cluster C_{new} and published it. Also, C_{new} is added to Set_{kc}.

Algorithm 2 $AnonymizationPBM(t)$

```
Find K-anonymous cluster C_min from Set_kc which covers t with minimum information loss
if C_min is found then
    Use cluster generalization of C_min to publish t
    RETURN
end if
Let P_t be the host partition of the expiring tuple t
Create temporary partition's set S_pub
Let QID_p be publication QID initialized as P_t.qid
for each P_i ∈ Set_p do
    if P_i.qid ⊆ QID_p then
        Add P_i to S_pub
    end if
end for
do
    Find the biggest partition P_sim that is the most similar to QID_p
    QID_p = QID_p ∪ P_sim.qid
    for each P_i ∈ Set_p do
        if P_i.qid ⊆ QID_p & P_i ∉ S_pub then
            Add P_i to S_pub
        end if
    end for
while (∑_{P_i∈S_pub} |P_i|) ≥ K
Find K − 1 nearest tuple of t from S_pub and form new cluster C_new
Publish C_new and remove tuples from partitions accordingly
Add C_new to Set_kc
```

$SuppressAnonymization(t)$ is called when data streams is ended or interrupted. It tries to find *K*-anonymous cluster C_{min} from Set_kc which covers *t* with minimum information loss (see definition 3). If such *K*-anonymous cluster is found, then it publishes *t* with the generalization information of C_{min}. Otherwise, tuple *t* is suppressed and published.

V. Experimental evaluation

To evaluate the performance of $IncrementalPBM$, we compared it with *K-VARP* [18]. The *K-VARP* is the most recent algorithm for anonymizing missing data streams. We

Algorithm 3 $SuppressAnonymization(t)$

1: Find K-anonymous cluster C_{min} from Set_{kc} which covers t' with minimum information loss
2: **if** C_{min} exists **then**
3: Use cluster generalization of C_{min} to publish t';
4: **else**
5: Suppress t' and publish
6: **end if**
7: Remove t' from P'

TABLE I
PARAMETERS OF ALGORITHMS

Algorithm	Parameters
K-VARP	K=50, δ=2000, ω=2000, R=0.2
IncrementalPBM	K=50, δ=2000, ω=2000

used famous dataset $Adult$[1] to evaluate the anonymization algorithms [14]–[16], [18], [27]. $Adult$ dataset has fourteen $QIDs$ containing eight categorical and six numeric attributes which are: *education, marital-status, work-class, occupation, relationship, race, gender, country* and *age, final-weight, education-number, capital-gain, capital-loss*. The generalization hierarchy of eight categorical attributes are defined in [13], and summary of $Adult$ dataset is described in [18]. We used 30000 tuples of $Adult$ dataset for the experiment, and missing values are added to imitate the missing data stream. To simulate consistent flow of data stream each tuple is retrieved from the dataset with the delay of 500ms. Both algorithms are implemented in *Java*. The experiments are performed on a PC with Intel® Core™ i7-6700HQ CPU with 16GB RAM and *Windows 10x64* with *JDK8.0*.

The parameter for the experiments shown in Table I, where K is the K-anonymity, δ is the time constraint of the sliding window, ω is the time constraint re-usable K-anonymous clusters and R is the R-likeness of $K-VARP$.

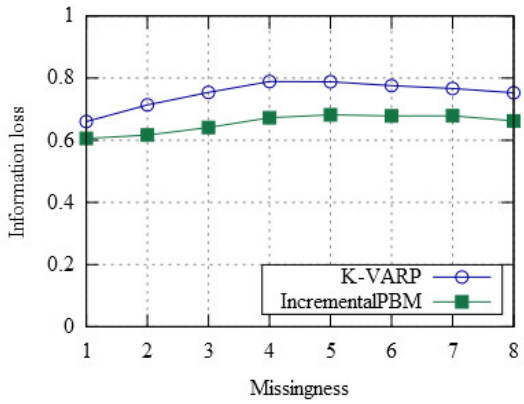

Fig. 1. Average information loss of *IncrementalPBM* and *K-VARP*

The average information loss on both *IncrementalPBM* and *K-VARP* is illustrated in Fig. 1. The average information loss is calculated using equation (5). Overall, *IncrementalPBM* demonstrated around 5% to 9% percent less information loss compared to *K-VARP*. Both algorithm showed increase of information loss when missingness of the data is increased from 1-missing to 4-missing. However, *IncrementalPBM* performent more consistent and efficient anonymization in terms of information loss. The less information of *IncrementalPBM* is caused by the incremental partitioning based marginalization which increased the number of tuples for *KNN* while minimizing the number of QIDs for anonymization.

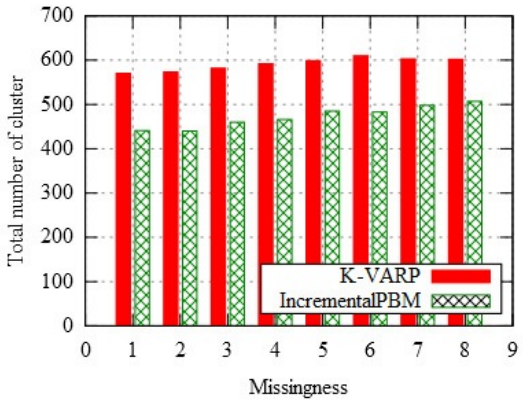

Fig. 2. Number of Clustering of *IncrementalPBM* and *K-VARP*

Fig. 2 demonstrated the number of clusters created for *IncrementalPBM* and *K-VARP*. *K-VARP* published approximately 30% more clusters compared to the *IncrementalPBM*. Nevertheless, we can see gradual increase on number of clusters when missingness of data is increasing. *K-VARP* utilizes less number of partitions for single anonymization iteration compared to *IncrementalPBM*. Therefore, less number of small partitions are appeared for *K-VARP* and this caused less number of clustering for *IncrementalPBM*.

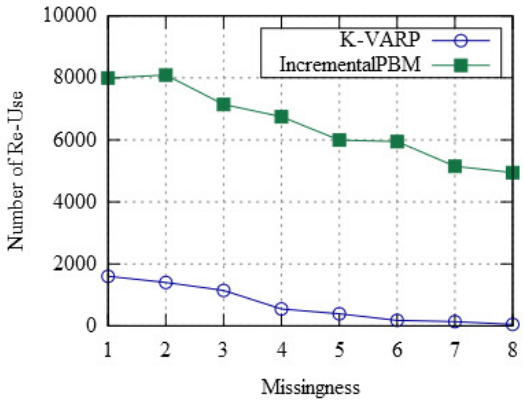

Fig. 3. Number of Re-Use of *IncrementalPBM* and *K-VARP*

The number of re-use anonymization of *IncrementalPBM* and *K-VARP* are shown in Fig. 3. The difference of re-use anonymization is very large, *IncrementalPBM* re-use anonymized more than 4500 to 6000 tuples compared to *K-*

[1]https://archive.ics.uci.edu/ml/datasets/adult

VARP. *IncrementalPBM* uses less time to prepare tuples for *KNN*. To merge partitions in *K-VARP* calculates distance of every tuples of partitions, in contrast, *IncrementalPBM* uses partitions' QID and size for partition accumulation. Also, *IncrementalPBM* prioritizes re-use before clustering, and this resulted big amount of re-use anonymization. Fig. 4 depicts the number of suppression for both *IncrementalPBM* and *K-VARP*. Number of suppressions for both algorithms are not substantial compared to the size of the test dataset. *K-VARP* suppressed one to fourteen tuples, whereas, *IncrementalPBM* suppressed one to six tuples. Substantial number of re-using occured for *IncrementalPBM*, and reasonably fit number of clustering occured on *K-VARP*. Therefore, minimal suppression is expected for the both algorithms.

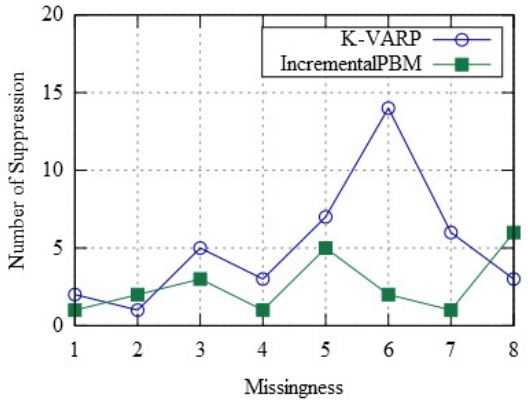

Fig. 4. Number of Suppression of *IncrementalPBM* and *K-VARP*

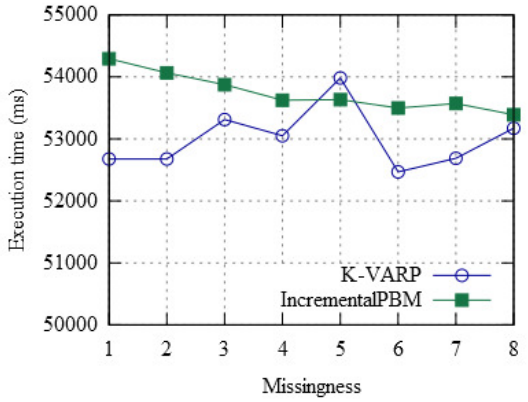

Fig. 5. Runtime of *IncrementalPBM* and *K-VARP*

The runtime of algorithms are illustrated on Fig. 5. Overall, *IncrementalPBM* showed very steady runtime for all the experiments, run time of *IncrementalPBM* is gradually decreased from 54400ms to 53400ms when missingess of the data is increased. On the other hand, *K-VARP* showed unstable runtime. Also, when missingness is less, then, *K-VARP* spent noticeable less time for anonymization compared *IncrementalPBM*. Overall, average runtime difference of both algorithm is approximately 500ms which is relatively short time compared to the total runtime.

VI. Conclusion

In this paper, we presented an algorithm *IncrementalPBM* for anonymizing missing data streams. *IncrementalPBM* can be utilized to process missing data streams generated from IoT environment. For example, condition monitoring of buildings, smart parking and smart homes. The incremental partitioning based marginalization helps to control the number of QIDs for anonymization while increasing the number of tuples for anonymization. Compared to existing algorithm *K-VARP*, *IncrementalPBM* showed considerable improvement, 5% to 9% less information loss, 4500 to 6000 more number of re-use anonymization while showing comparable number of clusters created, tuples suppressed, and runtime. For future work, we will concentrate on the optimization of clustering of missing values in IoT system data streams. Finally, we will work on the development of adaptive partitioning based marginalization algorithm for anonymizing missing data streams using artificial intelligence techniques.

Acknowledgement

The first author likes to acknowledge the support provided by the EU Erasmus Mundus project gLINK (552099-EM-1-2014-1-UK-ERA) to pursue this research at the University of the West of Scotland, UK.

References

[1] M. Oppitz and P. Tomsu, "Internet of things," in *Inventing the Cloud Century*. Springer, 2018, pp. 435–469.

[2] P. Krishnan, "Design of collision detection system for smart car using li-fi and ultrasonic sensor," *IEEE Transactions on Vehicular Technology*, 2018.

[3] W. Alsafery, B. Alturki, S. Reiff-Marganiec, and K. Jambi, "Smart car parking system solution for the internet of things in smart cities," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2018, pp. 1–5.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart community: an internet of things application," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[6] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, p. 13, 2016.

[7] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[8] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "\ell-diversity: Privacy beyond\kappa-anonymity," in *null*. IEEE, 2006, p. 24.

[9] T. M. Truta and B. Vinay, "Privacy protection: p-sensitive k-anonymity property," in *null*. IEEE, 2006, p. 94.

[10] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 106–115.

[11] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 49–60.

[12] X. Zhang, C. Liu, S. Nepal, and J. Chen, "An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 542–555, 2013.

[13] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 279–288.

[14] H. Zakerzadeh and S. L. Osborn, "Faanst: fast anonymizing algorithm for numerical streaming data," in *Data privacy management and autonomous spontaneous security*. Springer, 2011, pp. 36–50.

[15] K. Guo and Q. Zhang, "Fast clustering-based anonymization approaches with time constraints for data streams," *Knowledge-Based Systems*, vol. 46, pp. 95–108, 2013.

[16] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan, "Castle: Continuously anonymizing data streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 337–352, 2011.

[17] H. Zakerzadeh and S. L. Osborn, "Delay-sensitive approaches for anonymizing numerical streaming data," *International journal of information security*, vol. 12, no. 5, pp. 423–437, 2013.

[18] A. Otgonbayar, Z. Pervez, K. Dahal, and S. Eager, "K-varp: K-anonymity for varied data streams via partitioning," *Information Sciences*, vol. 467, pp. 238–255, 2018.

[19] J. Han, J. Yu, Y. Mo, J. Lu, and H. Liu, "Mage: A semantics retaining k-anonymization method for mixed data," *Knowledge-Based Systems*, vol. 55, pp. 75–86, 2014.

[20] A. Oganian and J. Domingo-Ferrer, "Local synthesis for disclosure limitation that satisfies probabilistic k-anonymity criterion." *Transactions on Data Privacy*, vol. 10, no. 1, pp. 61–81, 2017.

[21] S. Kim, M. K. Sung, and Y. D. Chung, "A framework to preserve the privacy of electronic health data streams," *Journal of biomedical informatics*, vol. 50, pp. 95–106, 2014.

[22] W. Wang, J. Li, C. Ai, and Y. Li, "Privacy protection on sliding window of data streams," in *Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on*. IEEE, 2007, pp. 213–221.

[23] J. Wang, C. Deng, and X. Li, "Two privacy-preserving approaches for publishing transactional data streams," *IEEE Access*, vol. 6, pp. 23 648–23 658, 2018.

[24] J. Li, B. C. Ooi, and W. Wang, "Anonymizing streaming data for privacy protection," 2008.

[25] K. Al-Hussaeni, B. C. Fung, and W. K. Cheung, "Privacy-preserving trajectory stream publishing," *Data & Knowledge Engineering*, vol. 94, pp. 89–109, 2014.

[26] M. Al-Zobbi, S. Shahrestani, and C. Ruan, "Experimenting sensitivity-based anonymization framework in apache spark," *Journal of Big Data*, vol. 5, no. 1, p. 38, 2018.

[27] P. Wang, J. Lu, L. Zhao, and J. Yang, "B-castle: An efficient publishing algorithm for k-anonymizing data streams," in *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, vol. 2. IEEE, 2010, pp. 132–136.

[28] A. Otgonbayar, Z. Pervez, and K. Dahal, "Toward anonymizing iot data streams via partitioning," in *Mobile Ad Hoc and Sensor Systems (MASS), 2016 IEEE 13th International Conference on*. IEEE, 2016, pp. 331–336.

[29] X. Yan, W. Xiong, L. Hu, F. Wang, and K. Zhao, "Missing value imputation based on gaussian mixture model for the internet of things," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[30] J. W. Graham, "Multiple imputation with norm 2.03," in *Missing data*. Springer, 2012, pp. 73–94.

[31] B. Fekade, T. Maksymyuk, M. Kyryk, and M. Jo, "Probabilistic recovery of incomplete sensed data in iot," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2282–2292, 2018.

[32] C.-F. Tsai, M.-L. Li, and W.-C. Lin, "A class center based approach for missing value imputation," *Knowledge-Based Systems*, vol. 151, pp. 124–135, 2018.

[33] Z.-g. Liu, Q. Pan, J. Dezert, and A. Martin, "Adaptive imputation of missing values for incomplete pattern classification," *Pattern Recognition*, vol. 52, pp. 85–95, 2016.

[34] K. Wagstaff, "Clustering with missing values: No imputation required," in *Classification, Clustering, and Data Mining Applications*. Springer, 2004, pp. 649–658.

[35] M. Ciglic, J. Eder, and C. Koncilia, "Anonymization of data sets with null values," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXIV*. Springer, 2016, pp. 193–220.

[36] A. B. Sakpere and A. V. Kayem, "On anonymizing streaming crime data: A solution approach for resource constrained environments," in *IFIP International Summer School on Privacy and Identity Management*. Springer, 2017, pp. 170–186.

[37] S. Li, "Poisson process with fuzzy rates," *Fuzzy Optimization and Decision Making*, vol. 9, no. 3, pp. 289–305, 2010.

[38] J. Barnard and X.-L. Meng, "Applications of multiple imputation in medical studies: from aids to nhanes," *Statistical methods in medical research*, vol. 8, no. 1, pp. 17–36, 1999.

[39] E.-L. Silva-Ramírez, R. Pino-Mejías, and M. López-Coello, "Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns," *Applied Soft Computing*, vol. 29, pp. 65–74, 2015.

[40] Z. Zhang, "Missing data imputation: focusing on single imputation," *Annals of translational medicine*, vol. 4, no. 1, 2016.

[41] P. Li, E. A. Stuart, and D. B. Allison, "Multiple imputation: a flexible tool for handling missing data," *Jama*, vol. 314, no. 18, pp. 1966–1967, 2015.

[42] S. Van Buuren, *Flexible imputation of missing data*. Chapman and Hall/CRC, 2018.

[43] S. Kosub, "A note on the triangle inequality for the jaccard distance," *Pattern Recognition Letters*, vol. 120, pp. 36–38, 2019.